AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-36. (Cancelled).

37. (Previously Presented) A method of evaluating a collection of data according to claim 76, wherein said step of determining a K context and its corresponding K context count value comprises the steps of:

inputting a selection to provide an input selection, said input selection containing a context constraint list having values represented by at least one root node of said interlocking trees data store to provide an input selection, wherein all of said nodes representing said context constraint list are associated with each other by a logical expression contained in said input selection;

identifying one or more K paths of said plurality of K paths within said interlocking trees datastore by their respective end product nodes from said at least one root node by traversing from an asResult link list of said at least one root node to said at least one root node's corresponding subcomponent node and traversing asCase links between said corresponding subcomponent node to each corresponding end product node of said subcomponent node;

disregarding those of the identified K paths that have links to elemental root nodes the value fields of which do not conform with said logical expression, a resultant store of nodes thus being a K context consisting of all those nodes along only those K paths which have not been

disregarded; and

adding said counts of the end product nodes of those one or more K paths which have not been disregarded to obtain said K context count.

38. (Previously Presented) A method of evaluating a collection of data according to claim 37, wherein said logical expression includes at least one logical operator selected from the following operators: AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO or any combination of such logical operators.

39. (Previously Presented) A method of evaluating a collection of data according to claim 76, wherein said step of determining said K context and its corresponding K context count value comprises the steps of:

inputting to said process a selection containing a Context constraint list containing values represented by at least one said root node of said interlocking trees data store wherein all of said at least one root nodes on said Context constraint list are associated with each other by a logical expression contained in said selection;

identifying one or more K paths by end product node by traversing from all possible end product nodes back toward the primary root using Case links along said K path, and, at each subcomponent node using its Result link to locate and compare the root node to said at least one root node;

disregarding those K paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a K context being nodes along only those K paths which have not been disregarded; and

adding the counts of the end product nodes of those one or more K paths, which have not been disregarded to obtain a K context count.

40. (Previously Presented) A method of evaluating a collection of data according to claim 39, wherein said logical expression includes at least one logical operator selected from the following operators: AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO or any combination of such logical operators.

41. (Previously Presented) A method of evaluating a collection of data according to claim 76, wherein said step of determining said K context and its corresponding context count value comprises the steps of:

selecting a plurality of K paths of said interlocking trees data store by end product node;

disregarding K paths of said plurality of K paths that have links to elemental root nodes, said value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a K context including nodes along only those K paths which have not been disregarded; and

adding said counts of said end product nodes of those one or more K paths of said plurality of K paths which have not been disregarded to obtain a K context count.

42. (Previously Presented) A method of evaluating a collection of data according to claim 76, wherein said step of determining said focus and its corresponding value comprises the steps of:

selecting a focus constraint list of at least one root node from the root nodes or the elemental root nodes of said interlocking trees data store, said at least one root node being

associated by a logical expression;

identifying one or more K paths by end product node from said at least one root node by traversing from the asResult list of the at least one root node to any corresponding subcomponent node and traversing said corresponding subcomponent node's asCase links to its corresponding end product node; to provide an established K context;

disregarding those K paths not within said established K context; and

disregarding those K paths that have links to elemental root nodes having value fields which do not conform to said logical expression, a resultant set of nodes thus forming a focus including nodes along only those K paths which have not been disregarded, and

adding said counts of said end product nodes of those one or more K paths which form said focus in order to obtain a focus count.

43. (Previously Presented) A method of evaluating a collection of data according to claim 42, wherein said logical expression includes at least one logical operator selected from the following: AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO or any combination of such logical operators.

44. (Previously Presented) A method of evaluating a collection of data according to claim 76, wherein said step of determining said focus and its corresponding value comprises the steps of:

selecting a focus constraint list of at least one root node from the root nodes or the elemental root nodes of said interlocking trees data store, said at least one root node being associated by a logical expression;

identifying one or more K paths by end product node by traversing from all end product nodes within established K context back along K paths toward their primary root nodes, said K paths identifiable using Case links of said end product nodes within said established K context, and while traversing, at each subcomponent node using the Result link to locate and compare the root node to said at least one root node;

disregarding those K paths that have links to elemental root nodes having value fields which do not conform to said logical expression, a resultant set of nodes thus forming a focus including nodes along only those K paths which have not been disregarded; and

adding said counts of said end product nodes of those one or more K paths which have not been disregarded to obtain a focus count.

45. (Previously Presented) A method of evaluating a collection of data according to claim 44, wherein said logical expression includes at least one logical operator selected from the following operators: AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO or any combination of such logical operators.

46. (Canceled)

47. (Previously Presented) A method of evaluating a collection of data according to claim 77, wherein the step of determining a position along each K path of the K context comprises the steps of:

selecting a root node from said root nodes or said elemental root nodes of said interlocking trees data store; and

traversing from said root node's or elemental root node's asResult list to its corresponding

subcomponent node in each K path of said K context.

48. (Previously Presented) A method of evaluating a collection of data according to claim 77, wherein said step of determining said K context and its corresponding K context count value comprises the steps of:

inputting to said process a selection which contains a Context constraint list containing values represented by at least one root node of said interlocking trees data store wherein all of the at least one root nodes on said Context constraint list are associated with each other by a logical expression contained within said selection;

identifying one or more K paths by end product node from said at least one root node by traversing from an asResult list of the at least one root node to the at least one root node's corresponding subcomponent node and traversing asCase links between said corresponding subcomponent node to each corresponding end product node of said subcomponent node;

disregarding those K paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a K context being nodes along only those K paths which have not been disregarded; and

adding the counts of the end product nodes of those one or more K paths which have not been disregarded to obtain a K context count.

49. (Previously Presented) A method of evaluating a collection of data according to claim 48, wherein said logical expression includes at least one logical operator selected from the following operators: AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO or any combination of such logical operators.

50. (Previously Presented) A method of evaluating a collection of data according to claim 49, wherein said step of determining said K context and its corresponding K context count value comprises the steps of:

inputting to said process a selection which contains a context constraint list containing values represented by at least one root node of said interlocking trees data store wherein all of said at least one root nodes on said context constraint list are associated with each other by a logical expression contained in said selection;

identifying one or more K paths by end product node by traversing from all possible end product nodes back toward the primary root using Case links along said K path and at each subcomponent node using its Result link to locate and compare the root node to said at least one root node;

disregarding those K paths that have links to elemental root nodes the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a K context being nodes along only those K paths which have not been disregarded; and

adding the counts of the end product nodes of those one or more K paths which have not been disregarded to obtain a K context count.

51. (Previously Presented) A method of evaluating a calculating a collection of data according to claim 50, wherein said logical expression includes at least one logical operator selected from the following operators: AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO or any combination of such logical operators.

52. (Previously Presented) A method of evaluating a collection of data according to

claim 77, wherein said step of determining said K context and its corresponding value comprises the steps of:

selecting all possible K paths of said interlocking trees data store by end product node;

disregarding those K paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression a resultant set of nodes thus forming a K context including nodes along only those K paths which have not been disregarded; and

adding said counts of said end product nodes of those one or more K paths which have not been disregarded to obtain a K context count.

53. (Previously Presented) A method of evaluating a collection of data according to claim 77, wherein said step of determining said focus and its corresponding value comprises the steps of:

selecting a focus constraint list of at least one root node from the root nodes or the elemental root nodes, of said interlocking trees data store, said at least one root node being associated by a logical expression;

identifying one or more K paths by end product node, from said at least one root node, by traversing from the asResult list of the at least one root node to any corresponding subcomponent node and traversing said corresponding subcomponent node's asCase links to its corresponding end product node.

disregarding those K paths not within the established K context; and

disregarding those K paths that have links to elemental root nodes having value fields which do not conform to said logical expression, a resultant set of nodes thus forming a focus

including nodes along only those K paths which have not been disregarded, and

adding said counts of said end product nodes of those one or more K paths which form said focus in order to obtain a focus count.

54. (Previously Presented) A method of evaluating a collection of data according to claim 53, wherein said logical expression includes at least one logical operator selected from the following operators: AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO or any combination of such logical operators.

55. (Previously Presented) A method of evaluating a collection of data according to claim 77, wherein said step of determining a focus and its corresponding value comprises the steps of:

selecting a focus constraint list of at least one root node from the root nodes or the elemental root nodes of said interlocking trees data store, said at least one root node being associated by a logical expression;

identifying one or more K paths by end product node, by traversing from all end product nodes within said established K context back along K paths toward their primary root nodes, said K paths identifiable using Case links of said end product nodes within established K context, and while traversing at each subcomponent node using the Result link to locate and compare the root node to said at least one root node;

disregarding those K paths that have links to elemental root nodes having value fields which do not conform to said logical expression, a resultant set of nodes thus forming a focus including nodes along only those K paths which have not been disregarded; and,

adding the counts of the end product nodes of those one or more K paths, which have not been disregarded to obtain a focus count.

56. (Previously Presented) A method of evaluating a collection of data according to claim 55, wherein said logical expression includes at least one logical operator selected from the following operators: AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO or any combination of such logical operators.

57. (Canceled).

58. (Previously presented) A method of evaluating a collection of data according to claim 78, wherein the step of determining said position along each K path of the K context comprises the steps of:

selecting a root node from the root nodes or the elemental root nodes of said interlocking trees data store and traversing from said root node's or elemental root node's asResult list to its corresponding subcomponent node in each K path of the K context.

59. (Previously presented) A method of evaluating a collection of data according to claim 78, wherein said step of determining a K context and its corresponding K context count value comprises the steps of:

inputting to said process a selection which contains a Context constraint list containing values represented by at least one root node of said interlocking trees data store, wherein all of the at least one root nodes on said Context constraint list are associated with each other by a logical expression contained within said selection;

identifying one or more K paths by end product node from said at least one root node by

traversing from an asResult list of said at least one root node to said at least one root node's corresponding subcomponent node and traversing asCase links between said corresponding subcomponent node to each corresponding end product node of said subcomponent node;

disregarding those K paths that have links to elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a K context being nodes along only those K paths which have not been disregarded; and

adding said counts of said end product nodes of those one or more K paths which have not been disregarded to obtain a K context count.

60. (Previously Presented) A method of evaluating a collection of data according to claim 59, wherein said logical expression includes at least one logical operator selected from the following operators: AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO or any combination of such logical operators.

61. (Previously presented) A method of evaluating a collection of data according to claim 78, wherein said step of determining said K context and its corresponding K context count value comprises the steps of:

inputting to said process a selection which contains a Context constraint list containing values represented by at least one root node of said interlocking trees data store, wherein all of the at least one root nodes on said Context constraint list are associated with each other by a logical expression contained within said selection;

identifying one or more K paths by end product node by traversing from all possible end product nodes back toward said primary root using Case links along said K path, and at

each subcomponent node using its Result link to locate and compare the root node to said at least one root node;

disregarding those K paths that have links to said elemental root nodes, the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a K context being nodes along only those K paths which have not been disregarded; and

adding said counts of said end product nodes of those one or more K paths which have not been disregarded to obtain a K context count.

62. (Previously Presented) A method of evaluating a collection of data according to claim 61, wherein said logical expression includes at least one logical operator selected from the following operators: AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO or any combination of such logical operators.

63. (Previously presented) A method of evaluating a collection of data according to claim 78, wherein said step of determining said K context and its corresponding value comprises the steps of:

selecting a plurality of K paths of said interlocking trees data store by end product node;

disregarding those K paths that have links to elemental root nodes the value fields of which do not conform with said logical expression, a resultant set of nodes thus forming a K context including nodes along only those K paths which have not been disregarded; and

adding said counts of said end product nodes of those one or more K paths which have not been disregarded to obtain a K context count.

64. (Previously presented) A method of evaluating a collection of data according to claim

78, wherein said step of determining said focus and its corresponding value comprises the steps of:

selecting a focus constraint list of at least one root node, from the root nodes or the elemental root nodes, of said interlocking trees data store, said at least one root node being associated by a logical expression;

identifying one or more K paths by end product node from said at least one root node by traversing from said asResult list of said at least one root node to any corresponding subcomponent node and traversing said corresponding subcomponent node's asCase links to its corresponding end product node to provide an established context;

disregarding those K paths not within said established K context; and

disregarding those K paths that have links to elemental root nodes having value fields which do not conform to said logical expression, a resultant set of nodes thus forming a focus including nodes along only those K paths which have not been disregarded, and

adding said counts of the said product nodes of those one or more K paths which form said focus in order to obtain a focus count.

65. (Previously Presented) A method of evaluating a collection of data according to claim 64, wherein said logical expression includes at least one logical operator selected from the following operators: AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO or any combination of such logical operators.

66. (Previously presented) A method of evaluating a collection of data according to claim 78, wherein said step of determining said focus and its corresponding value comprises the

steps of:

selecting a focus constraint list of at least one root node from the root nodes or the elemental root nodes of said interlocking trees data store, said at least one root node being associated by a logical expression;

identifying one or more K paths by end product node by traversing from all end product nodes within said established K context back along K paths toward their primary root nodes, said K paths identifiable using Case links of said end product nodes within established K context and while traversing, at each subcomponent node using the Result link to locate and compare the root node to said at least one root node;

disregarding those K paths that have links to elemental root nodes having value fields which do not conform to said logical expression, a resultant set of nodes thus forming a focus including nodes along only those K paths which have not been disregarded; and

adding the counts of the end product nodes of those one or more K paths, which have not been disregarded to obtain a focus count.

67. (Previously Presented) A method of evaluating a collection of data according to claim 66, wherein said logical expression includes at least one logical operator selected from the following operators: AND, OR, and NOT, GREATERTHAN, LESSTHAN, XNOR, EQUALTO or any combination of such logical operators.

68. (Previously Presented) A structure for providing a useful arrangement of information relating to predetermined values is stored within a memory and accessible by a computer, said structure comprising nodes and links between said nodes, each of said nodes having a plurality of

data fields, at least two data fields of said plurality of data fields containing a pointer, one of said at least two pointers being a Case pointer and the other of said at least two pointers being a Result pointer and at least one node having at least one additional pointer to a list of pointers, one of said additional pointers to said list of pointers being to an asCase list of pointers in instances where said node has an associated asCase list and another being to asResult list of pointers in instances where said node has associated an associated asResult list, and wherein said nodes contain a count field, and wherein said nodes include types of nodes called root nodes of which there are at least one primary root node and at least one elemental root node and wherein said nodes may include other root nodes, said nodes further including types of nodes called end of thought nodes of which there is in said structure at least one end of thought node, types of nodes called subcomponent nodes of which there is in said structure at least one subcomponent node, and types of nodes called end product nodes of which there is in said structure at least one end of thought node, types of nodes called subcomponent nodes of which there is in said structure at least one subcomponent node, and types of nodes called end product node of which there is in said structure at least one of thought node, and wherein said asResult links point between a said root node and any other of said node types, and wherein said asCase links point between said at least one primary root node and said at least one end product node, include including in a K path therebetween at least one subcomponent node and wherein said elemental root nodes also have a field having a one of said values.

69. (Previously Presented) The structure of claim 68 wherein said structure is formed from a set of program instructions which configure a computer system when activated therein to

produce said structure, responsive to the presentation of information to said set of program instructions.

70. (Previously Presented) A computer readable medium containing the set of program instructions as set forth in claim 69.

71. (Previously Presented) The structure set forth in claim 68 wherein said count field contains an intensity variable, said intensity variable modifiable at various intensities corresponding to various predetermined traversal types of activity related to a node containing said count field.

72. (Previously Presented) structure as set forth in claim 68 wherein said asCase and said asResult lists are stored in a separate data structure from said interlocking trees structure and wherein said separate data structure is associated with related nodes in said interlocking trees structure by pointers.

73. (Previously Presented) A structure for providing a useful arrangement of information relating to predetermined values is stored within a memory and accessible by a computer, said structure comprising nodes and links between said nodes, each of said nodes having a plurality of data fields, at least two of said plurality of data fields containing a pointer, one of said at least two pointers being a Case pointer and the other of said at least two pointers being a Result pointer and at least one node having at least one additional pointer to a list of pointers, one of said additional pointers to said list of pointers being to an asCase list of pointers in instances where said node has associated asCase list and another being to an asResult list in instances where said node has associated an asResult list of pointers, and wherein said nodes are provided

with one sub-node for each predetermined manner of traversal, said sub-nodes containing a count field for recording traversals of said nodes in predetermined manners, and wherein said nodes include types of nodes called root nodes of which there are at least one primary root node and at least one elemental root node and wherein said nodes may include other root nodes, said nodes further including types of nodes called end of thought nodes of which there is in said structure at least one end of thought node, types of nodes called subcomponent nodes of which there is in said structure at least one subcomponent node, and types of nodes called end product nodes of which there is in said structure at least one end product node, and wherein said asResult links point between a said root node and any other of said node types, and wherein said asCase links point between said at least one primary root node and said at least one end product node, including in a K path therebetween at least one subcomponent node and wherein said elemental root nodes also have a field having a one of said values.

74. (Previously Presented) A structure for providing a useful arrangement of information relating to predetermined values is stored within a memory and accessible by a computer, said structure comprising nodes and links between said nodes, said nodes having a plurality of data fields, at least two of said plurality of data fields containing a pointer, one of said at least two pointers being a Case pointer and the other of said at least two pointers being a Result pointer and at least one node having at least one additional pointer to a list of pointers, one of said additional pointers to said list of pointers being to an asCase list of pointers in instances where said node has associated asCase list and another being to asResult list of pointers in instances where said node has associated an asResult list, and wherein said nodes contain an additional

field, and wherein said nodes include a type of node called root nodes of which there are at least one primary root node and at least one elemental root node and wherein said nodes may include other root nodes, said nodes further including other types of nodes including at least one end of thought node, at least one subcomponent node, and at least one end product node, and wherein said asResult links point between said root node and any other of said node types, and wherein said asCase links point between said at least one primary root node and said at least one end product node, including in a K path therebetween at least one subcomponent node and wherein said elemental nodes also have a field having a one of said values.

75. (Previously Presented) The structure of claim 74 wherein said additional field is a count field.

76. (Previously Presented) A method of evaluating a collection of data represented by an interlocking trees data store situated within active memory accessible to a process running in a computer, said interlocking trees datastore comprising a plurality of K paths having a structured collection of nodes connected by links of said nodes having pointers to other nodes of said interlocking trees datastore, wherein said nodes contain a count field, said nodes including at least nominally different kind of nodes, a first kind called root nodes of which there are at least one primary root node and at least one elemental root node and which may include other root nodes, a second kind of node called an end of thought node, at least one node of a third kind of node called a subcomponent node, and at least one node of a kind of node called an end product node, and wherein there exist at least two kinds of said links, asResult and asCase links, wherein said asResult links point between a one of said root nodes and any other node, and wherein said

asCase links point between said at least one primary root node and at least one said end product node and include in a K path between said end product node and said primary root node at least one said subcomponent node said method comprising the steps of:

traversing at least one K path of said plurality of K paths using at least one link of said asResult links or said asCase links;

determining a K context within said data store in accordance with said traversing of said at least one K path;

determining a corresponding context value of said K context;

determining a focus within said K context and its focus corresponding focus value;

calculating a probability of an occurrence of said focus within said K context in accordance with said corresponding K context value and said focus value; and

providing a probability value corresponding to said probability of said occurrence of said focus within said K context to said process running in said computer.

77. (Previously Presented) A method of evaluating a collection of data represented by an interlocking trees data store situated within active memory accessible to a process running in a computer, said interlocking trees datastore comprising a plurality of K paths having a structured collection of nodes connected by links of said nodes having pointers to other nodes of said interlocking trees datastore, wherein said nodes contain a count field, said nodes including at least nominally different kinds of nodes, a first kind called root nodes of which there are at least one primary root node and at least one elemental root node and which may include other root nodes, a second kind of node called an end of thought node, at least one node of a third kind of

node called a subcomponent node, and at least one node of a kind of node called an end product node, and wherein there exist at least two kinds of said links, asResult and asCase links, wherein said asResult links point between a one of said root nodes and any other node, and wherein said asCase links point between said at least one primary root node and at least one said end product node and include in a K path between said end product node and said primary root node at least one said subcomponent node said method comprising the steps of:

traversing at least one K path of said plurality of K paths using at least one link of said asResult links or said asCase links;

determining a K context within said data store in accordance with said traversing of said at least one K path;

determining a position along each K path of said K context;

determining a focus within said K context and its corresponding focus value; calculating a probability of an occurrence of said focus between said position and said end product node along at least one K path within said K context; and providing said probability of said occurrence of said focus between said position and the end product along the K path within said K context to said process running in said computer.

78. (Previously Presented) A method of evaluating a collection of data represented by an interlocking trees data store situated within active memory accessible to a process running in a computer, said interlocking trees datastore comprising a plurality of K paths having a structural collection of nodes connected by links of said nodes having pointers to other nodes of said interlocking trees datastore, wherein said nodes contain a count field, said nodes including at

least nominally different kinds of nodes, a first kind called root nodes of which there are at least one primary root node and at least one elemental root node and which may include other root nodes, a second kind of node called an end of thought node, at least one node of a third kind of node called a subcomponent node, and at least one node of a kind of node called an end product node, and wherein there exist at least two kinds of said links, asResult and asCase links, wherein said asResult links point between a one of said root nodes and any other node, and wherein said asCase links point between said at least one primary root node and at least one said end product node and include in a K path between said end product node and said primary root node at least one said subcomponent node, said method comprising the steps of:

traversing at least one K path of said plurality of K paths using at least link of said asResult links or said asCase links;

determining a K context within said data store in accordance with said traversing of said at least one K path and determining its corresponding context value;

determining a position along each K path of the K context;

determining a focus within said K context and its corresponding value;

calculating the a probability of the occurrence of said focus between said position and the primary root, along the K path within said K context; and

providing said probability of an occurrence of said focus between said position and the primary root along said K path within said K context to said process running in said computer.